University of Lisbon

ISEG, Lisbon School of Economics and Management

MSc in Data Analytics for Business

Programming for Data Science

Academic year 2021/2022

What are the factors that make an Airbnb accommodation more attractive?

Group Project

Authors:

Rodrigo Conceição	52528
João Silvestre	50719
Rosanna Mueller	56280
Marc Behse	56246

Table of contents

List of figures

1 Introduction

Airbnb has a significant impact on the tourism in Lisbon. Since 2010 the number of Airbnb accommodation is increasing constantly (Amore, A., de Bernardi, C., Arvanitis, P., 2020). Currently, there are a total of 18.527 Airbnb accommodations located in Lisbon (Inside Airbnb, 2022).

The guests can choose between different type of accommodations, in different municipalities and key words, such as “super host”, should indicate the guests which hosts are popular. In report, we want to identify factors that really make a difference. What are the factors that make an Airbnb accommodation more attractive?

This study seeks to explore this question in further detail. Our objective is to create value for Airbnb hosts who want to offer an attractive accommodation in order to increase their revenue. Furthermore, we want to deliver insights to Airbnb, since the company’s revenue model is based on commissions from successful bookings.

First, a literature review will consolidate the theoretical background related to the subject. Next, the methodology applied for the project in the area of Data Science will be described. In chapter 4 the results of the data analysis and of the use of different algorithms will be explained. Moreover, the outcomes are evaluated in this section. In the final chapter, the key results are summarized, and the empirical findings are linked with our research objective.

2 Literature review

First, the project group did a literature review and hat a look at already existing studies on the subject. Several studies that analyzed the Airbnb listings dataset could be found. However, the project group noticed that most of the studies focus on Airbnb listings in the United States.

For example, in January 2019 the Data Scientist Sarang Gupta published on the platform “Towards Data Science” an exploratory analysis of the Airbnb Listings Dataset for New York. The study aims to understand the rental landscape in New York through various static and interactive visualizations. For this purpose, among other factors, the Airbnb’s host network, the number of ratings and the type of accommodations were analyzed. The analysis highlights a few trends from data to give an overview of Airbnb’s market. For example, Sarang Gupta found out that most Airbnb reveal high ratings and high response rates, but only a small fraction becomes a superhost. Moreover, January tends to be the quietest month in New York and the occupancy rate increases throughout the year (Towards Data Science, 2019).

Another interesting study is published in October 2021 on the website “Analytics Vidhya”. It is a predictive analysis on Airbnb listings data of Seattle. The author of the study followed a method called CRISP-DM. The predictive analysis reveals some key findings which can be informative for hosts because they can gain better ratings or higher prices. However, there are also some key findings for tourists, such as the best time to visit or easy commute options (Analytics Vidhya, 2021).

3 Methodology

In order to achieve a good quality of the work the project group decided to choose the CRISP-DM (Cross-Industry Standard Process for Data Mining) methodology. This approach is created in 1996 and provides a complete data mining process, which enables everyone to understand and follow the several steps for the implementation of a data mining project. The CRISP-DM life cycle consists of six phases, which are illustrated in the figure 1 below (Costa, Aparicio, 2020).

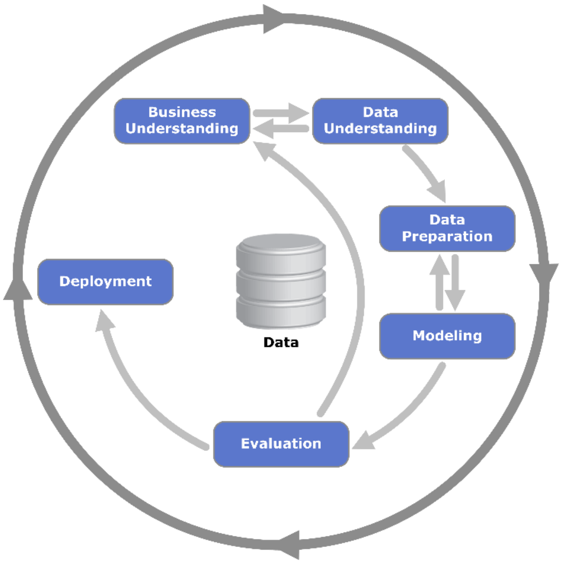


Figure 1: CRISP-DM Lifecyle (Tribloom , 2019)

In the following the phases will be described by showing how the project group implemented them into practice.

Business Understanding:

The first phase is understanding the company. It focuses on understanding the project goals and requirements from a business perspective (Costa, Aparicio, 2020).

In this stage the group first analyzed the Airbnb business model. Founded in 2008 Airbnb became a global brand that is used by millions of people around the world. At its heart, it is a platform business model that connects hosts with people who are looking for accommodation. The platform is fully digital; there is no face-to-face service or physical store where the clients need to go. Moreover, Airbnb does not own any property itself. Airbnb only enables to connect people who have an accommodation to offer with people who are interested in finding an accommodation. For each booking Airbnb receives commissions from two sources: hosts and guests. For each booking, Airbnb charges the guest 6-12% of the booking fee. Airbnb also charges the host 3% for each successful transaction (Business Model Toolbox, 2022).

Furthermore the project group defined the main question which should be answered through the analysis. The question is called: “What are the factors that make an Airbnb accommodation more attractive?”. The Airbnb hosts were identified as the target group of this analysis, since they want to offer an attractive accommodation to increase revenue. Moreover, Airbnb belongs to the target group as the company’s revenues model is based on commissions from successful bookings.

Lastly, the project group designed a preliminary plan to achieve the objectives. This plan contained the main idea of the implemented methods:

- Measure attractiveness through the number of positive reviews
- Sentiment Analysis based on reviews
- Regression Analysis to predict if apartment would get a good review given a set of regressors
- Network Analysis to show the most important Airbnb hosts

Data Understanding:

The data understanding phase consists of three primary steps: data collection, data properties, and data quality (Tribloom 2019).

The project group decided to work with the following datasets:

Data	Source	Description
Listings data	Inside Airbnb, 2022	Detailed Listings data for Lisbon
Calendar data	Inside Airbnb, 2022	Booking data for Lisbon from 08.12.21
Reviews data	Inside Airbnb, 2022	Review Data for listings in Lisbon
Neighborhoods and Geographic data	Inside Airbnb, 2022	Neighbourhood list and Geo Information
Crime data	Pordata, 2020	Crime-Rate per municipality
Population Density	Pordata, 2020	Population Density by Municipality
Purchasing power	Pordata, 2020	Purchasing power per Capita per municipality
Attractiveness of location	Autoridade tributária e aduaneira, 2022	Proximity to Lisbon City Centre, Proximity to Coast

Next, the group continues with activities to familiarize oneself with the data. For example, data dictionaries which describe the meaning of the data fields in a dataset helped to get a deeper understanding of the data. Moreover, data quality issues were identified and approaches to solve them were defined.

Data Preparation:

The data preparation phase includes all activities up to the creation of the final data set. At the end of this phase, the data used by the modelling tools is available (Costa, Aparicio, 2020).

In this phase the project group mainly cleaned the following AirBnB data files: Listings data, Reviews data and Neighborhoods data. For example, reviews without content (e. g. "-", ":)") were deleted in the Reviews data and unnecessary columns removed in the Listings data. Furthermore, the neighborhood data was merged with the crime rate, coefficient of location, purchasing power and population density.

Modelling:

In this phase, various modelling techniques are selected and applied. Also their parameters are adjusted to optimal values (Costa, Aparicio, 2020).

The project group decided to use several models to understand the data better and gain insights of interest. For example, a Sentiment analysis, a multivariate linear regression model, but also Machine Learning algoritms like Random Forest or K-means clustering were applied. The results of the varios algorithms are described in detail in the following chapter.

During the modelling process the project group noticed that additional data could optimize the performance of models (for example the linear regression model). Therefore, the group decided to go back to the data preparation phase and add data (purchasing power per capita, population density, number of crimes per thousand inhabitants and coefficient of location) to the neighborhoods data frame.

Evaluation:

In the evaluation, the performance of the models in relation to the business objectives defined in phase 1 "Business Understanding" are assessed and and a decision of the usage of the model is made (Tribloom 2019).

After implementation of various algorithms, the project group evaluated the performance of the selected models. The identification of limitations and the proposal of further research were part of this phase. The performance evaluation of the applied algorithms will be described in the following chapter. Furthermore, in the “Conclusions” chapter the results will be summarized and related to the business objectives.

4 Results

In this chapter the project group will explain the main results of the project. The results can be divided into two parts: Descriptive Statistics and inductive statistics.

4.1 Descriptive Statistics

In this chapter the project group performed descriptive statistics, mainly to understand the data better. First, general descriptive statistics will be applied. Then, geospatial analysis will be implemented and finally a sentiment analysis will be carried out.

4.1.1 General Descriptive Analysis

In this chapter, the project group performed some descriptive statistics in order to understand the data better and to reveal some key findings. First, the listings dataset is analyzed.

The bar chart below shows the top 10 hosts with most listings. On the x-axis the identification number of the host and on the y-axis the number of listings is displayed. It is visible that the number of listings of the top 10 hosts ranges from approximately 34 to 68 listings.

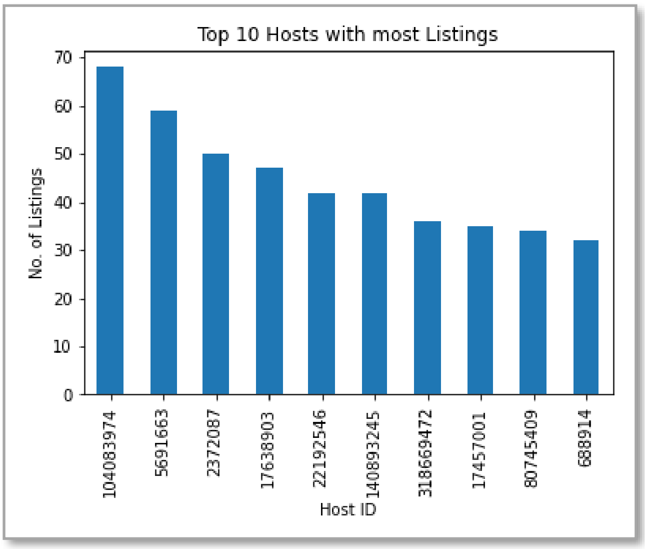


Figure 2: Top 10 Hosts with most Listings

The horizontal bar chart below presents the 5 neighborhoods with the highest number of listings. On the x-axis the number of listings and on the y-axis the name of the neighborhood is shown. The key finding is that for the neighborhoods “Santa Maria Maior”, “Misericrdia”, “Arroios, “Santo Antnio and “Cascais e Estoril” the most Airbnb listings are provided.

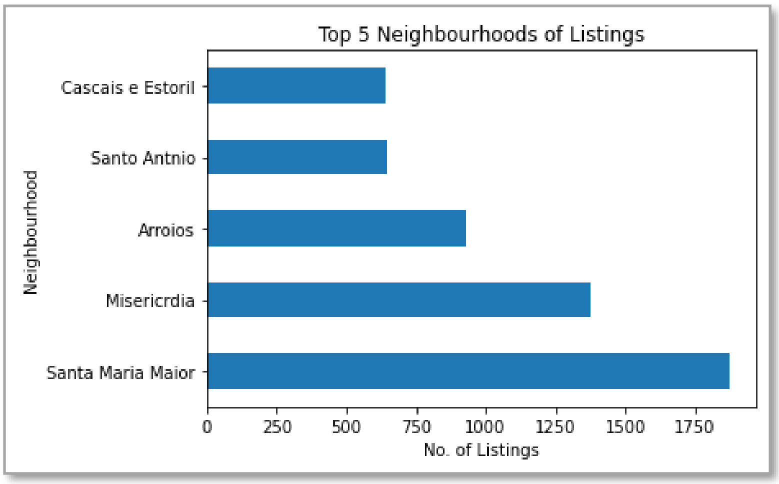


Figure 3: Top 5 Neighbourhoods of Listings

Next, a pie chart for the maximum capacity of the listings was created. It shows that most listings have a maximum capacity of 2, 4 or 6 persons. This is followed by a maximum capacity of 3 or 5 persons. Just a small fraction of listings allows a capacity of 7, 8, 9 or 10 persons. Also, a maximum capacity of one person is rarely.

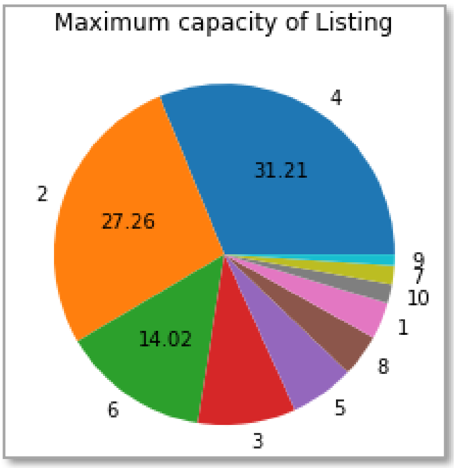


Figure 4: Maximum capacity of Listing

The pie chart below presents the number of listings with an average rating above 4 compared to those equal to 4 or below 4. It is visible that 93.5 % of the listings received an average rating score, which is higher than 4. This is an interesting finding and leads to the conclusion that most guests tend to give very good ratings.

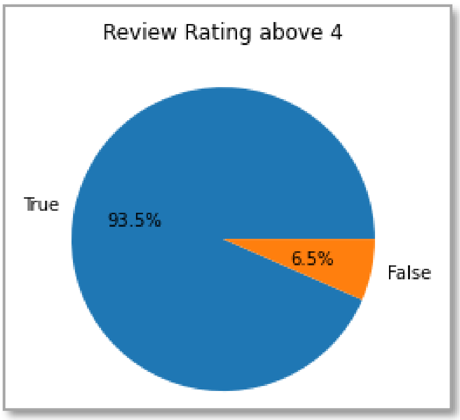


Figure 5: Review Rating above 4

The boxplot below analyzes the price of the listings. In order to enable an unbiased analysis most of the outliers were eliminated first. The plot shows that the median of the price is approximately 60 Euro. Moreover, 50 % of the listing's prices range between approximately 50 Euro and 90 Euro. The minimum price (excluding any outliers) is approximately 20 Euro and the maximum price 160 Euro (excluding any outliers).

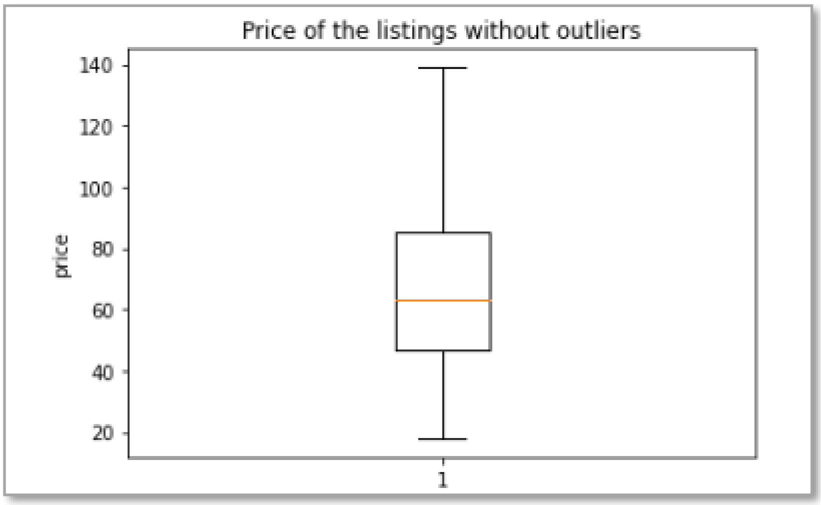


Figure 6: Price of listings with outliers

The following bar chart compares the number of super hosts with normal hosts. On the x-axis the type of host is displayed (0 = normal host, 1 = super host) and on the y-axis the number of hosts is displayed. There are approximately 6.500 normal hosts in Lisbon and 3.500 super hosts. This shows that around 30 % of the hosts in Lisbon are super hosts.

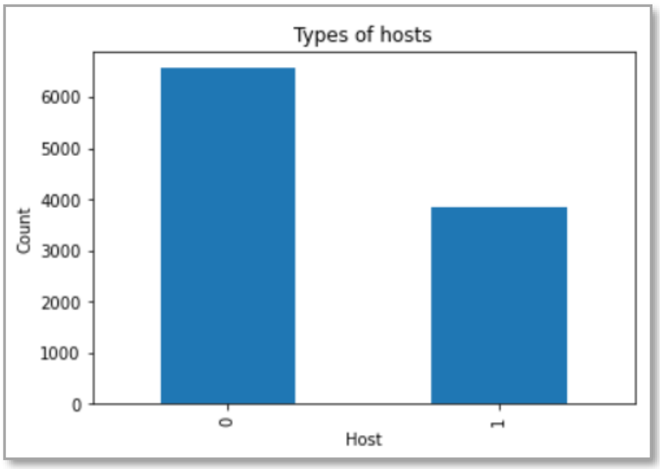


Figure 7: Types of hosts

Next, the number of bedrooms of the listings were analyzed and a pie chart for the visual representation is chosen. More than half of the listings provide just one bedroom, around 29 % of the listings 2 bedrooms and 11 percent 3 bedrooms. Listings with 4 or 5 bedrooms can rarely be found in Lisbon.

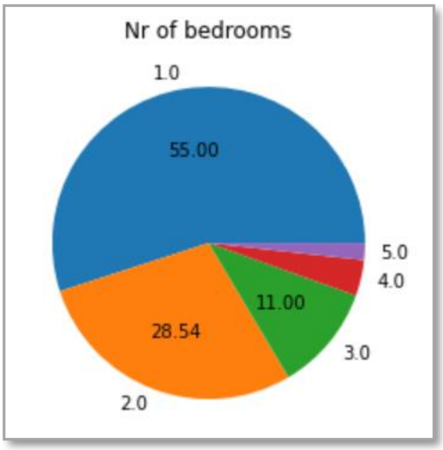


Figure 8: Number of bedrooms

4.1.2 Geospatial Analysis

4.1.2.1 Introduction to Geospatial analysis in Python

For the analysis of the Airbnb Data, the leafmap python package was used. This package contains nine modules for geomapping. One of those modules, the foliumap allows to plot backend upon the folium Python package (Wu, 2021). By relying on folium, we were able to visualize apartment prices, as well as host characteristics in an interactive way.

To map the coordinates, we were using shape files from OpenStreetMap. This platform follows a similar peer production model like Wikipedia. The overall objective of OpenStreetMap is to create an accurate set of map data that's free to use and easily available (Haklay and Weber, 2008).

Unlike other digital peer productions, in which individuals are creating most of the content with regional independence, the OpenStreetMap community organizes workshops for specific regions. Those workshops aim to fill localized geographical gaps with consistent data (Haklay and Weber, 2008). In practice, folium.Map has a direct connection the OpenStreetMap API. Using this API makes fetching and saving raw geodata files simple and efficient.

4.1.2.2 Geo mapping with Airbnb data

To initialize the map, a centre location must be defined. Our approach was to define the centre location by calculating the mean of all coordinates available. In Figure 9 we depicted the initialized map of all Airbnb apartments in the region of Lisbon with black circle markers. On this plot, we can see a high density of apartments in the city of Lisbon, as well as in Cascais and other touristic areas along the cost-side. The more rural areas in the centre of the map have a lower apartment density.

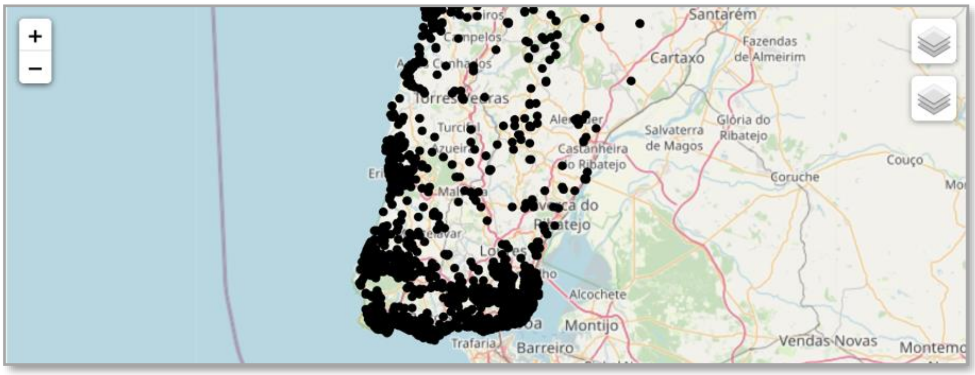


Figure 9: FoliumMap -All airbnb apartments

In Figure 10 the 1% of the most expensive airbnb apartments are mapped with blue folium markers. Surprisingly, there was no visual pattern between the closeness to the city centre and the price of the apartments.

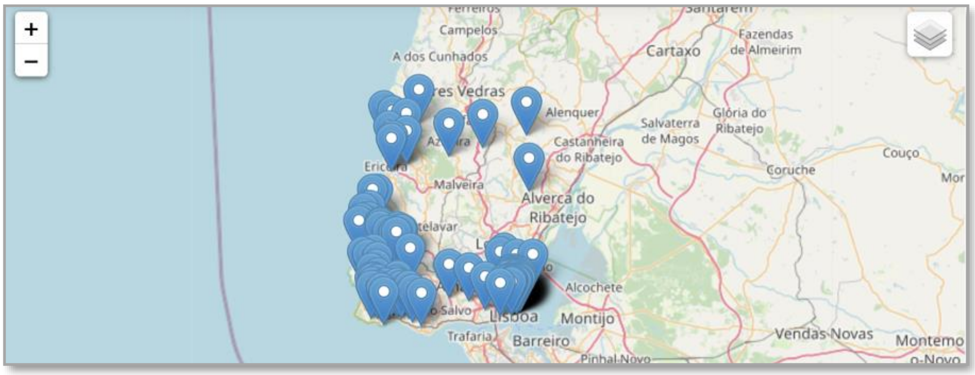


Figure 10: FoliumMap - expensive airbnb apartments

In Figure 11 all housings offered by the host with the highest number of apartments are mapped with black folium circles. The most prosperous host in Lisbon offers 68 housings which are mainly based in the centre of Lisbon with one exception in Terrugem and some apartments close to the airport.

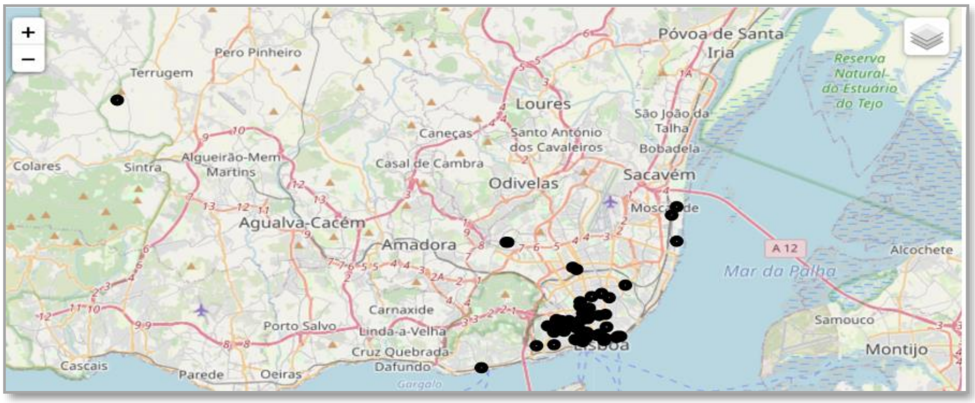


Figure 11: FoliumMap – Airbnb host with the highest number of apartments

In Figure 12 all housings offered by superhosts are mapped with indigo folium circles, non Superhosts plotted in slategrey. There is no regional pattern visible.

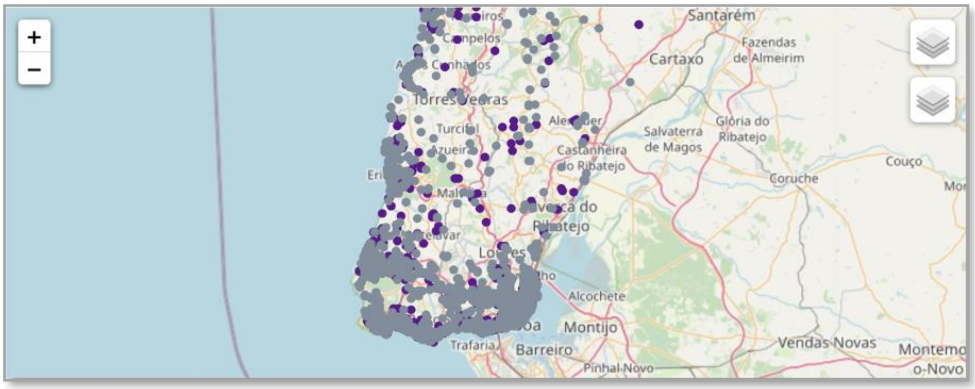
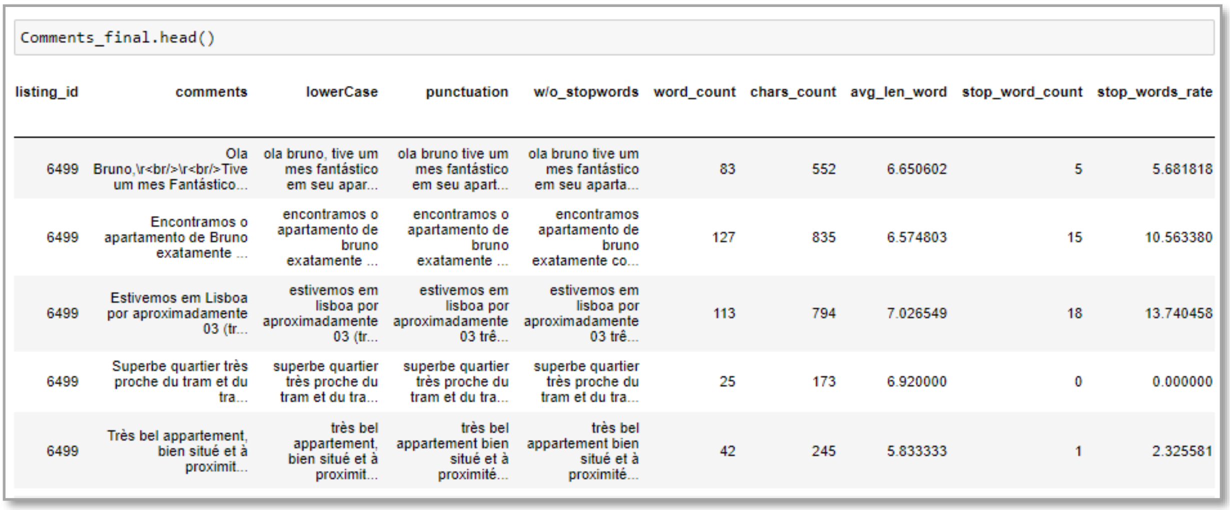


Figure 12: FoliumMap – superhosts vs. regular host

4.1.3 Sentiment Analysis

In this section, the objective was to analyze the comments from the listings and try to extract value from them. The group managed to extract some overall information (statistics) about the structure of the comments and created a new feature based on these comments that could be used in regression and classification models.

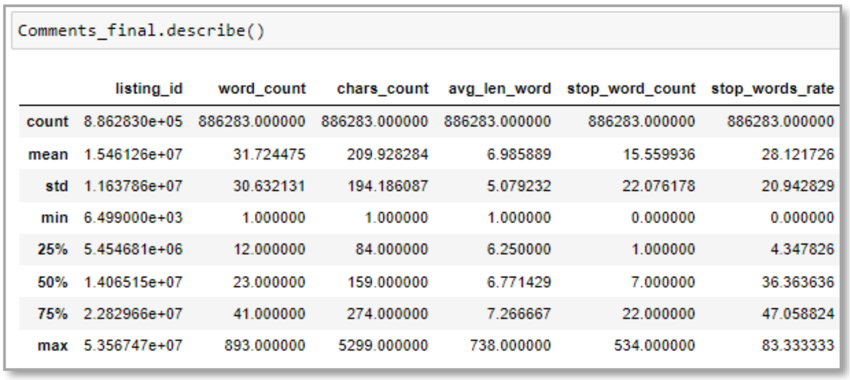
Before proceeding with the actual analysis, it was necessary to first clean the comments, namely transforming them into lowercase, removing punctuation and stop words. This cleaning was done in order to delete irrelevant information that could affect the comments analysis and to increase the efficiency of the the analysis. Each step done was appended to the comments data frame as a column so we could compare the progress done in each step and visualize it at the same time.



```
Comments_final.head()
```

listing_id	comments	lowerCase	punctuation	w/o_stopwords	word_count	chars_count	avg_len_word	stop_word_count	stop_words_rate
6499	Ola Bruno,\r<br/>\r<br/>Tive um mes Fantástico...	ola bruno, tive um mes fantástico em seu apar...	ola bruno tive um mes fantástico em seu apart...	ola bruno tive um mes fantástico em seu aparta...	83	552	6.650602	5	5.681818
6499	Encontramos o apartamento de Bruno exatamente ...	encontramos o apartamento de bruno exatamente ...	encontramos o apartamento de bruno exatamente ...	encontramos apartamento de bruno exatamente co...	127	835	6.574803	15	10.563380
6499	Estivemos em Lisboa por aproximadamente 03 (tr...	estivemos em lisboa por aproximadamente 03 (tr...	estivemos em lisboa por aproximadamente 03 trê...	estivemos em lisboa por aproximadamente 03 trê...	113	794	7.026549	18	13.740458
6499	Superbe quartier très proche du tram et du tra...	superbe quartier très proche du tram et du tra...	superbe quartier très proche du tram et du tra...	superbe quartier très proche du tram et du tra...	25	173	6.920000	0	0.000000
6499	Très bel appartement, bien situé et à proximit...	très bel appartement, bien situé et à proximit...	très bel appartement bien situé et à proximité...	très bel appartement bien situé et à proximité...	42	245	5.833333	1	2.325581

Figure 13: First 5 rows of the dataset

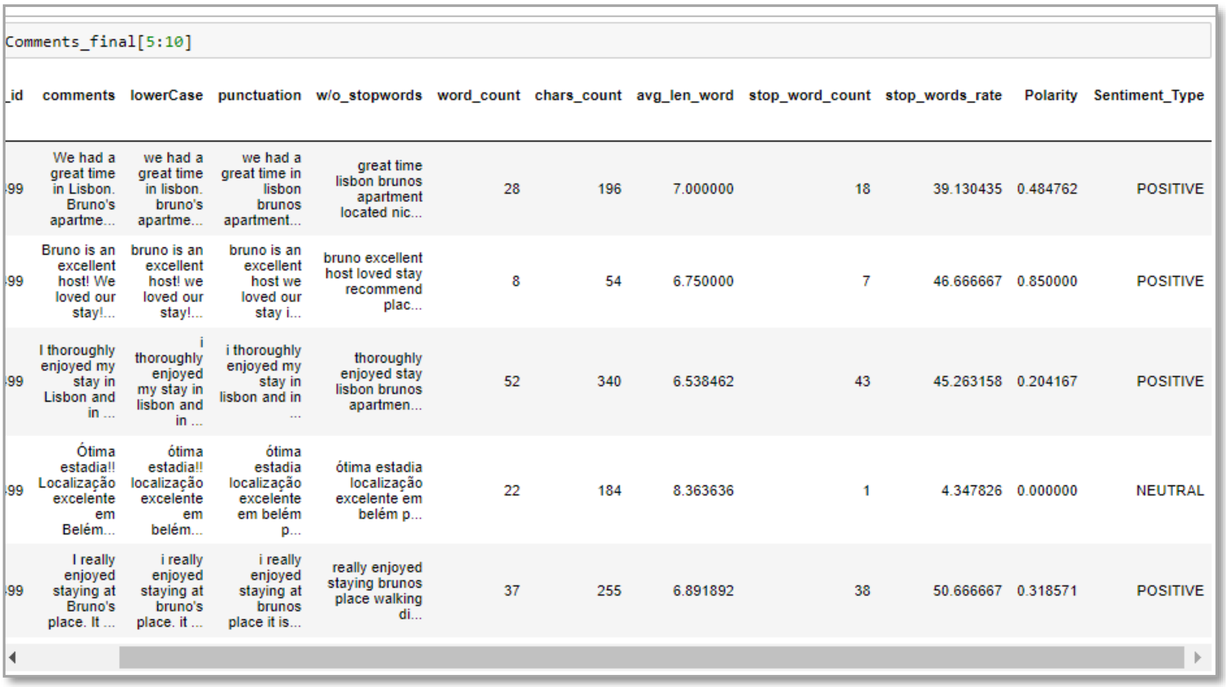


```
Comments_final.describe()
```

	listing_id	word_count	chars_count	avg_len_word	stop_word_count	stop_words_rate
count	8.862830e+05	886283.000000	886283.000000	886283.000000	886283.000000	886283.000000
mean	1.546126e+07	31.724475	209.928284	6.985889	15.559936	28.121726
std	1.163786e+07	30.632131	194.186087	5.079232	22.076178	20.942829
min	6.499000e+03	1.000000	1.000000	1.000000	0.000000	0.000000
25%	5.454681e+06	12.000000	84.000000	6.250000	1.000000	4.347826
50%	1.406515e+07	23.000000	159.000000	6.771429	7.000000	36.363636
75%	2.282966e+07	41.000000	274.000000	7.266667	22.000000	47.058824
max	5.356747e+07	893.000000	5299.000000	738.000000	534.000000	83.333333

Figure 14: Summary statistics of the variables

Regarding the comments structure, five new columns were appended to the comments data frame, they provide information for each comment about the number of words, number of characters, the average length of the words the number of stop words and the stop words rate. This allows the analyst to fastly understand if the comments contain many stop words or if they are well structured. Not only it provides an overall view but can also provide a grouped by vision for listings or an analysis for a single review.

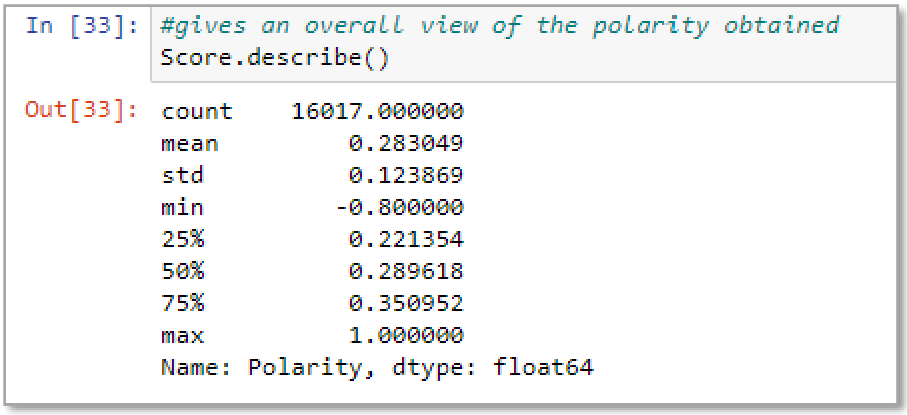


```
Comments_final[5:10]
```

_id	comments	lowerCase	punctuation	w/o_stopwords	word_count	chars_count	avg_len_word	stop_word_count	stop_words_rate	Polarity	Sentiment_Type
99	We had a great time in Lisbon. Bruno's apartme...	we had a great time in lisbon. bruno's apartme...	we had a great time in lisbon brunos apartment...	great time lisbon brunos apartment located nic...	28	196	7.000000	18	39.130435	0.484762	POSITIVE
99	Bruno is an excellent host! We loved our stay!...	bruno is an excellent host! we loved our stay!...	bruno is an excellent host we loved our stay i...	bruno excellent host loved stay recommend plac...	8	54	6.750000	7	46.666667	0.850000	POSITIVE
99	I thoroughly enjoyed my stay in Lisbon and in ...	i thoroughly enjoyed my stay in lisbon and in ...	i thoroughly enjoyed my stay in lisbon and in ...	thoroughly enjoyed stay lisbon brunos apartmen...	52	340	6.538462	43	45.263158	0.204167	POSITIVE
99	Ótima estadia!! Localização excelente em Belém...	ótima estadia!! localização excelente em belém...	ótima estadia localização excelente em belém p...	ótima estadia localização excelente em belém p...	22	184	8.363636	1	4.347826	0.000000	NEUTRAL
99	I really enjoyed staying at Bruno's place. It ...	i really enjoyed staying at bruno's place. it ...	i really enjoyed staying at brunos place it is...	really enjoyed staying brunos place walking di...	37	255	6.891892	38	50.666667	0.318571	POSITIVE

Figure 15: Final comments dataframe

To be able to use the comments as a feature in our models, it was necessary to calculate the polarity of each comment and append it to the data frame. This is a way of measure if the reviewers were satisfied with the listing through the comments and allows the analysts to understand the same three perspectives explained above, overall, listings and a particular review. It was also appended one more column regarding the sentiment type, this column classify the polarity with a score of positive, neutral or negative.

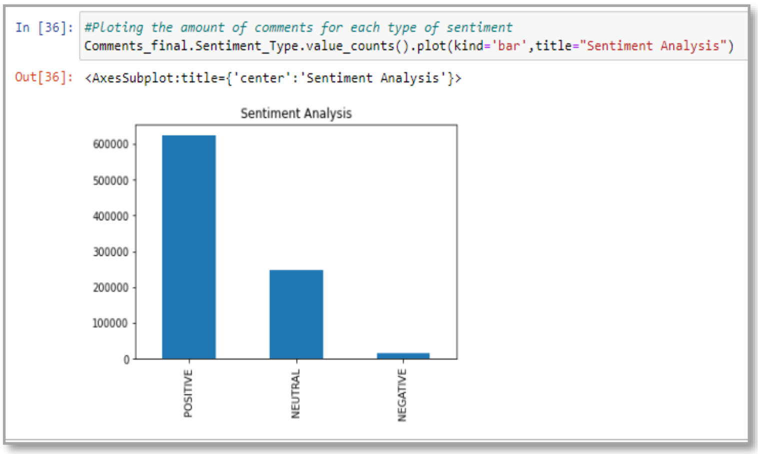


```
In [33]: #gives an overall view of the polarity obtained
         Score.describe()

Out[33]: count    16017.000000
         mean         0.283049
         std          0.123869
         min         -0.800000
         25%          0.221354
         50%          0.289618
         75%          0.350952
         max          1.000000
         Name: Polarity, dtype: float64
```

Figure 16: Summary statistics of the polarity

The next step was to group by the polarity by listing so we could merge it with the listings file and match with the other features.

apartment
clean
really nice
apartment
great
très bien
enjoyed stay
great host
location perfect
place great
stay
lisbon
bairro alto
walking distance
definitely recommend
très agréable
everything needed
great location
great place
highly recommend
come back
très propre
recommend place
everything need
perfect location
gave u
well located
appartement
bien situé
place stay
well equipped
highly recommended



```
In [36]: #Ploting the amount of comments for each type of sentiment
         Comments_final.Sentiment_Type.value_counts().plot(kind='bar',title="Sentiment Analysis")

Out[36]: <AxesSubplot:title={'center':'Sentiment Analysis'}>
```

Figure 17: Word cloud and sentiment type

Lastly, some visualizations were created to give the group an overall view of the data and some extra knowledge regarding the data. The word cloud provides us with a good image of the most frequent words on all the comments analyzed while the polarity bar plot gives intel about the overall classification of the comments. Looking at the bar plot, it's possible to notice that the overall of the comments are positive, which match the most frequent words, such as, great, perfect, recommend, which are associated to good reviews.

4.2 Inductive Statistics

In this chapter, our objective is to test hypotheses and make predictions. First, a hypothesis on the difference in two population is tested. Subsequently, the results of the sentiment analysis will be used to leverage the performance of the multilinear regression. In the last step, a clustering and calssification algorithm will be applied.

4.2.1 Hypothesis Testing

Hypothesis:

H0: The rental prices of apartments belonging to a superhost are lower than the apartments belonging to a non superhost or there is no difference in prices.

$\mu 1 \leq \mu 2$

H1: The rental prices of apartments belonging to a superhost are higher than the apartments belonging to a non superhost.

$\mu 1 > \mu 2$

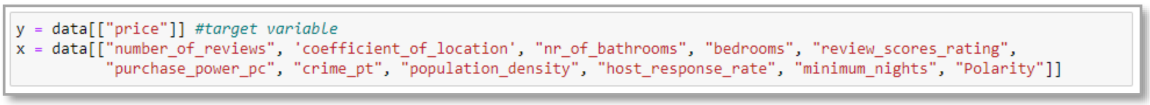
Differences for μ can be tested applying a t-test with alternative “greater”. However, it is assumed that both variables tested are normally distributed. The normal distribution assumption was tested with a Shapiro Wilk Test. The test had a significant result. Hence there is significant evidence that the variables are not normally distributed. An appropriate alternative for the t-test is the Mann-Whitney-U Test. This test can be applied on non-normally distributed data (McKnight and Najab, 2010).

The result of the test is a p-value=7.9342e-13. As the p-value is < 0.05, the H0 can be rejected and the H1 can be accepted. The rental prices of apartments belonging to superhosts are higher than the apartments belonging to a non superhosts.

The insight on this hypothesis is that the status as a superhost can be a factor of influence that makes an accommodation more attractive and hence more expensive.

4.2.2 Multivariate Linear Regression

The main goal of applying a multivariate linear regression model was to see how the target variable “price” was influenced by some of the key variables in the listings data set, as well as the other variables (external data) that were gathered and merged in the neighborhood’s csv. The result of the definition of the target variable and its predictors can be seen below:



```
y = data[["price"]] #target variable
x = data[["number_of_reviews", 'coefficient_of_location', "nr_of_bathrooms", "bedrooms", "review_scores_rating",
          "purchase_power_pc", "crime_pt", "population_density", "host_response_rate", "minimum_nights", "Polarity"]]
```

Figure 18: Allocation of Variables

Note that “Polarity” is the only variable which resulted from the sentiment analysis that was previously discussed. As a result, similar to the neighborhood’s csv, the polarity score given in sentiment analysis was also appended to each of its corresponding listings. Following this, the scatterplots between the target variable “price” and each given regressor was plotted:

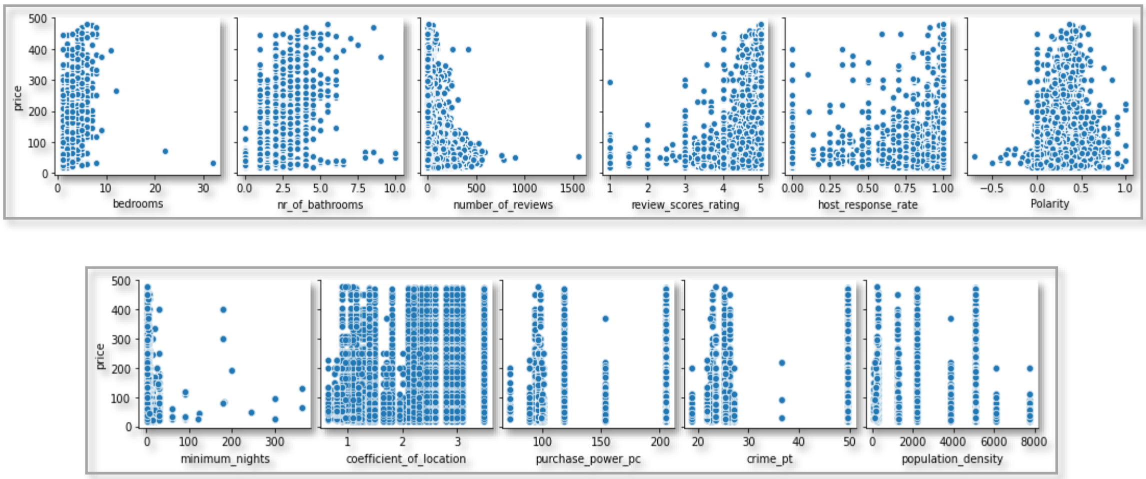
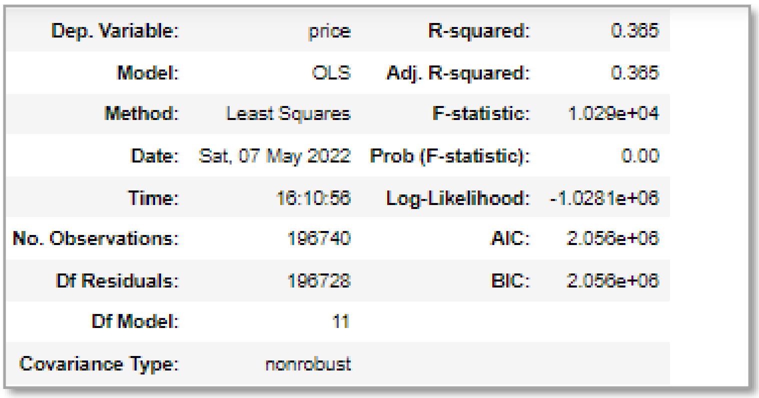


Figure 19: Scatterplots – Dependent and Independent Variables

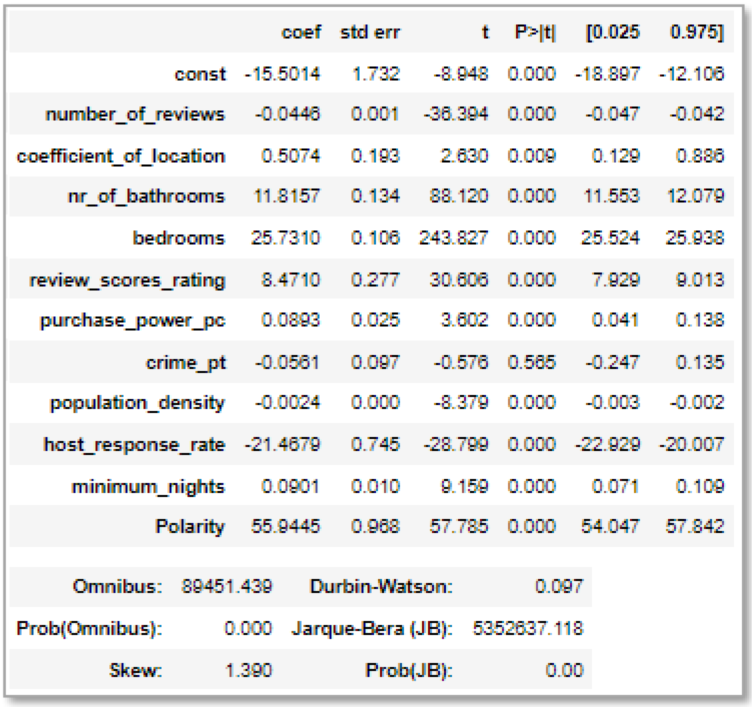
Naturally, not all the relationships seem to be linear. In fact, “host response rate” and “review scores rating” seem to have a similar exponential relationship with the variable “Price”.



Dep. Variable:	price	R-squared:	0.365
Model:	OLS	Adj. R-squared:	0.365
Method:	Least Squares	F-statistic:	1.029e+04
Date:	Sat, 07 May 2022	Prob (F-statistic):	0.00
Time:	16:10:56	Log-Likelihood:	-1.0281e+06
No. Observations:	196740	AIC:	2.056e+06
Df Residuals:	196728	BIC:	2.056e+06
Df Model:	11		
Covariance Type:	nonrobust		

Figure 20: General Output – Multivariate Linear Regression

As expected, after fitting an OLS model to the data, the R-squared value is 0.365. This is not an extremely low value, but it is not sufficiently high either. This means that only 36.5% of the variation in the variable “price” is explained by the variation in all the regressors given in “x”. Besides that, the AIC and BIC are extremely high numbers, which means that there is a very high probability of multicollinearity between the specified regressors.



	coef	std err	t	P>|t|	[0.025	0.975]
const	-15.5014	1.732	-8.948	0.000	-18.897	-12.106
number_of_reviews	-0.0446	0.001	-36.394	0.000	-0.047	-0.042
coefficient_of_location	0.5074	0.193	2.630	0.009	0.129	0.886
nr_of_bathrooms	11.8157	0.134	88.120	0.000	11.553	12.079
bedrooms	25.7310	0.106	243.827	0.000	25.524	25.938
review_scores_rating	8.4710	0.277	30.606	0.000	7.929	9.013
purchase_power_pc	0.0893	0.025	3.602	0.000	0.041	0.138
crime_pt	-0.0561	0.097	-0.576	0.565	-0.247	0.135
population_density	-0.0024	0.000	-8.379	0.000	-0.003	-0.002
host_response_rate	-21.4679	0.745	-28.799	0.000	-22.929	-20.007
minimum_nights	0.0901	0.010	9.159	0.000	0.071	0.109
Polarity	55.9445	0.968	57.785	0.000	54.047	57.842

Omnibus:	89451.439	Durbin-Watson:	0.097
Prob(Omnibus):	0.000	Jarque-Bera (JB):	5352637.118
Skew:	1.390	Prob(JB):	0.00

Figure 21: Coefficient Statistics – Multivariate Linear Regression

The presumption of multicollinearity is further verified by the given coefficient values. Although the p-values are significant for nearly all regressors, the vast majority of them have such small coefficients that they could very well be constants. Number of bathrooms, bedrooms, review scores rating, and polarity all seem to have a fairly high coefficient as well as significant p-values, indicating that the majority of the success of our model comes from these variables. By analyzing these coefficients, we can conclude that:

- Increasing the number of bathrooms by 1 leads to an increase in price by 11.81 euros
- Increasing the number of bedrooms by 1 leads to an increase in price by 25.73 euros
- Increasing the review scores rating by 1 leads to an increase in price by 8.47 euros
- Increasing the polarity score by 1 leads to an increase in price by 55.94 euros

Out of all of these regressors, the polarity score is the one with the highest coefficient. This is mainly due to the fact that polarity varies very little. Since polarity scores range from -1 to 1, increasing polarity by 1 is a very significant increase which corresponds to an overall extremely positive reviews of the listing, which consequently affects its price. Lastly, it is important to highlight that the external variables (crime per thousand inhabitants, purchase power per capita, population density and coefficient of location) seem to have no effect on price.

The distribution of the residuals for the multivariate linear regression seem to indicate that the OLS assumption that errors are normally distributed is not met either. The left tail of the distribution is smaller than the right tail of the distribution and, therefore, not normally distributed:

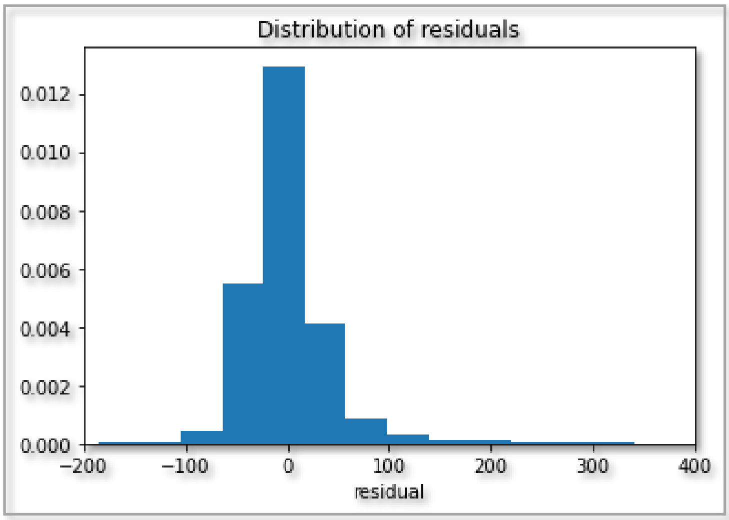
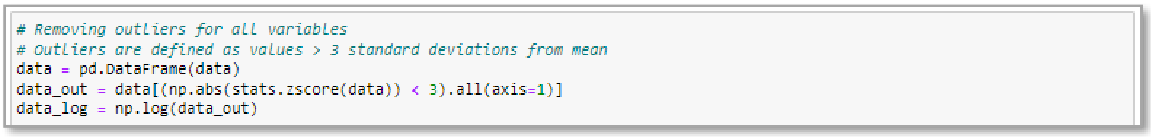


Figure 22: Histogram- Distribution of Residuals

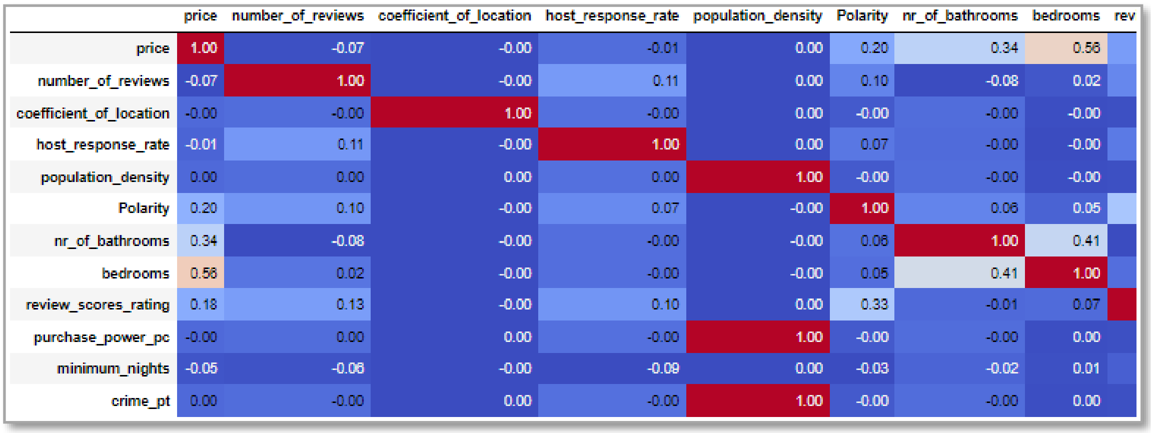
Given all these issues, an improvement of the multivariate linear regression was clearly necessary. As a result, the first step to remove the outliers and log-transform the data was taken:



```
# Removing outliers for all variables
# Outliers are defined as values > 3 standard deviations from mean
data = pd.DataFrame(data)
data_out = data[(np.abs(stats.zscore(data)) < 3).all(axis=1)]
data_log = np.log(data_out)
```

Figure 23: Transformation of variables

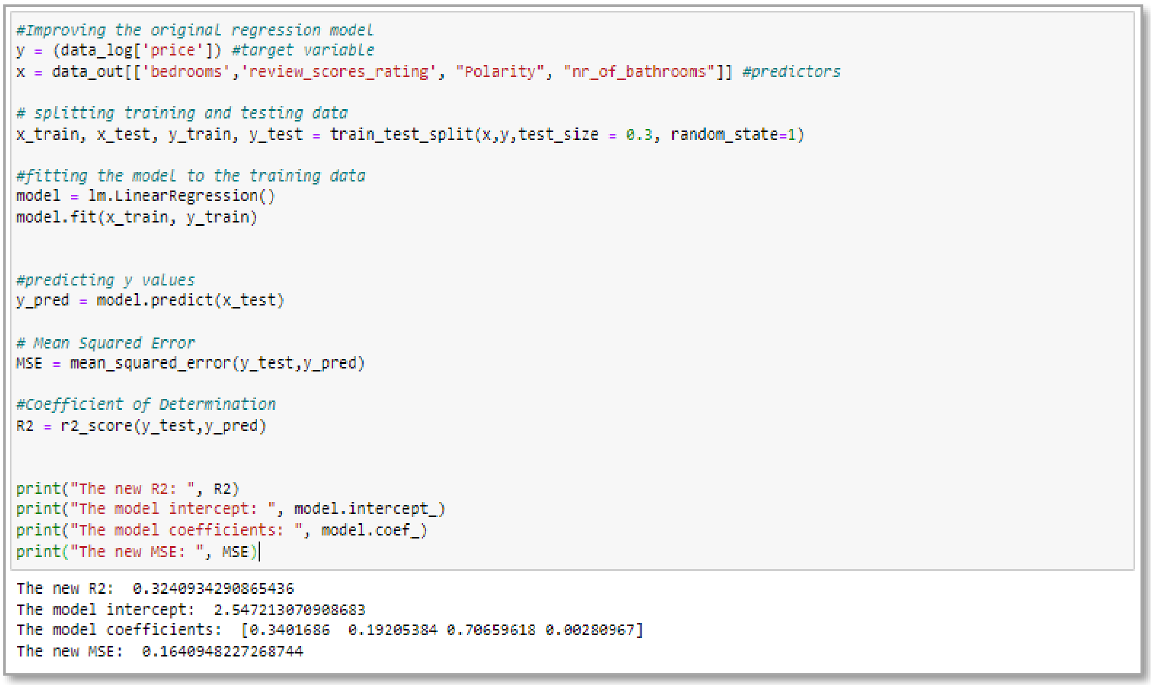
Secondly, a correlation matrix was computed in order to identify and select the regressors that had a significant correlation with price:



	price	number_of_reviews	coefficient_of_location	host_response_rate	population_density	Polarity	nr_of_bathrooms	bedrooms	rev
price	1.00	-0.07	-0.00	-0.01	0.00	0.20	0.34	0.56	
number_of_reviews	-0.07	1.00	-0.00	0.11	0.00	0.10	-0.08	0.02	
coefficient_of_location	-0.00	-0.00	1.00	-0.00	0.00	-0.00	-0.00	-0.00	
host_response_rate	-0.01	0.11	-0.00	1.00	0.00	0.07	-0.00	-0.00	
population_density	0.00	0.00	0.00	0.00	1.00	-0.00	-0.00	-0.00	
Polarity	0.20	0.10	-0.00	0.07	-0.00	1.00	0.06	0.05	
nr_of_bathrooms	0.34	-0.08	-0.00	-0.00	-0.00	0.06	1.00	0.41	
bedrooms	0.56	0.02	-0.00	-0.00	-0.00	0.05	0.41	1.00	
review_scores_rating	0.18	0.13	-0.00	0.10	0.00	0.33	-0.01	0.07	
purchase_power_pc	-0.00	0.00	0.00	-0.00	1.00	-0.00	-0.00	0.00	
minimum_nights	-0.05	-0.06	-0.00	-0.09	0.00	-0.03	-0.02	0.01	
crime_pt	0.00	-0.00	0.00	-0.00	1.00	-0.00	-0.00	0.00	

Figure 24: Correlation Matrix – Depended and Independent Variables

As can be seen in the correlation matrix, the variables bedrooms, bathrooms, review scores rating, and polarity are the ones that have a significant correlation with the variable price. As a result, a new multivariate regression model is built with these regressors:



```
#Improving the original regression model
y = (data_log['price']) #target variable
x = data_out[['bedrooms','review_scores_rating', "Polarity", "nr_of_bathrooms"]] #predictors

# splitting training and testing data
x_train, x_test, y_train, y_test = train_test_split(x,y,test_size = 0.3, random_state=1)

#fitting the model to the training data
model = lm.LinearRegression()
model.fit(x_train, y_train)


#predicting y values
y_pred = model.predict(x_test)

# Mean Squared Error
MSE = mean_squared_error(y_test,y_pred)

#Coefficient of Determination
R2 = r2_score(y_test,y_pred)


print("The new R2: ", R2)
print("The model intercept: ", model.intercept_)
print("The model coefficients: ", model.coef_)
print("The new MSE: ", MSE)
```

```
The new R2:  0.3240934290865436
The model intercept:  2.547213070908683
The model coefficients:  [0.3401686  0.19205384 0.70659618 0.00280967]
The new MSE:  0.1640948227268744
```

Figure 25: Output – Refined Multivariate Linear Regression

This time, the outliers were all removed, and the target variable “price” was log-transformed to allow for an easier interpretation of the model. The number of regressors were chosen in accordance with the correlation matrix and the data was split into training (70%) and testing data (30%). This time around, the obtained R-squared was 0.32. Although this R-squared is a bit lower than the one previously mentioned, the fact is that in this given model there are significantly less regressors. The coefficients can be interpreted as:

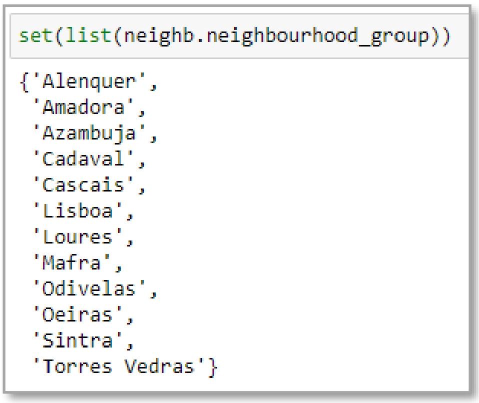
- If number of bedrooms increases by 1 unit, the price increases by 34% on average
- If review scores rating increases by 1 unit, the price increases by 19% on average
- If number of bathrooms increases by 1 unit, the price increases by 2.8% on average
- If polarity increases by 1 unit, the price increases by 70% on average

Interestingly enough, although the number of bathrooms had a significant correlation with price in the correlation matrix, its coefficient is rather small when compared to the other three. As was seen before, polarity varies very little, which is why an increase in polarity by 1 corresponds to such a high increase in price.

Either way, the multivariate linear regression analysis permitted the identification of the three main variables that influence the price of a given Airbnb listing in the city of Lisbon: the number of bedrooms the listing has, its review score and the polarity of the reviews given by the clients.

4.2.3 Clustering

The approach used when performing clustering with the given data is quite different from the other types of machine learning algorithms throughout this report. The intention here is to see how the clustering algorithm behaves given two specific variables and, after observing its results, try to extract a conclusion from it. Firstly, notice that the Airbnb dataset identifies listings from 12 different municipalities:



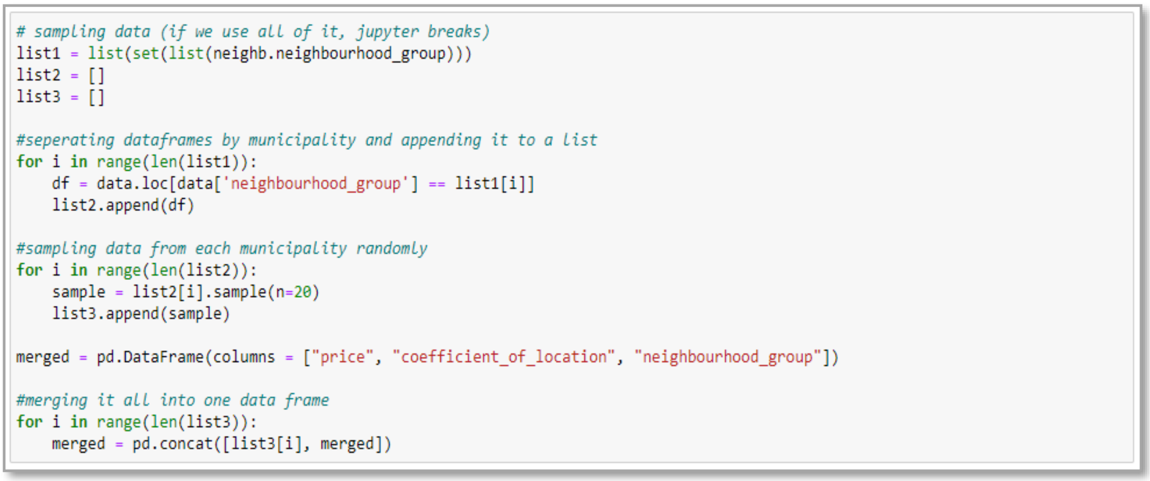
```
set(list(neighb.neighbourhood_group))
```

```
{'Alenquer',
 'Amadora',
 'Azambuja',
 'Cadaval',
 'Cascais',
 'Lisboa',
 'Loures',
 'Mafra',
 'Odivelas',
 'Oeiras',
 'Sintra',
 'Torres Vedras'}
```

Figure 26: List of Municipalities in the Region of Lisbon

Therefore, by applying k-means clustering to two given variables (price and coefficient of location), will the algorithm be able to identify the 12 municipalities? It is probably quite unlikely, due to the similarities between some of the data points for each municipality (the coefficient of location in Cascais are overall similar to the ones in Lisbon, for example). Either way, it will be possible to identify which municipalities the algorithm is grouping together.

Firstly, since the clustering algorithm is k-means and there is a substantial amount of data, it is important to sample the data, otherwise the algorithm would take too long to compute. Besides that, some municipalities have more listings than others, which forces the use of data balancing techniques for sampling:

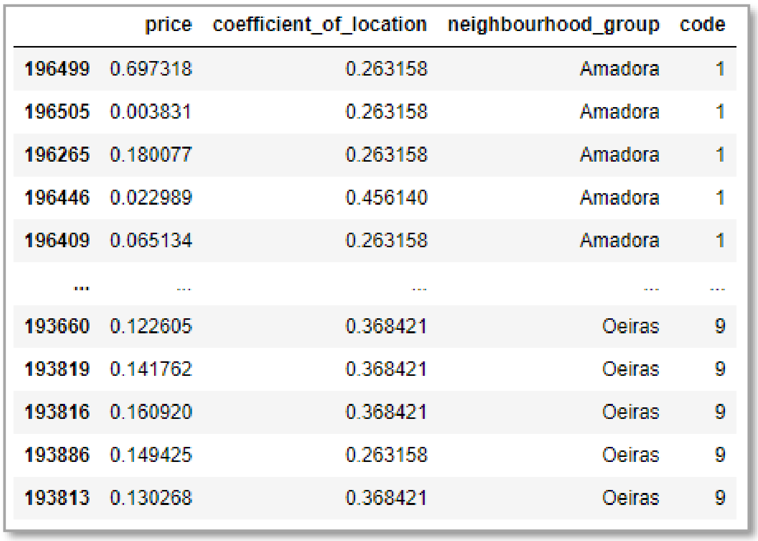


```
# sampling data (if we use all of it, jupyter breaks)
list1 = list(set(list(neighb.neighbourhood_group)))
list2 = []
list3 = []

#seperating dataframes by municipality and appending it to a list
for i in range(len(list1)):
    df = data.loc[data['neighbourhood_group'] == list1[i]]
    list2.append(df)

#sampling data from each municipality randomly
for i in range(len(list2)):
    sample = list2[i].sample(n=20)
    list3.append(sample)

merged = pd.DataFrame(columns = ["price", "coefficient_of_location", "neighbourhood_group"])

#merging it all into one data frame
for i in range(len(list3)):
    merged = pd.concat([list3[i], merged])
```

Figure 27: Sampling Process

The loops above implement an “undersampling” data balancing technique. Essentially, this ensures that each one of the 12 municipalities have exactly 20 observations in the data frame. This makes it so that the municipalities with more observations are brought down to an equivalent level as the ones with fewer observations.

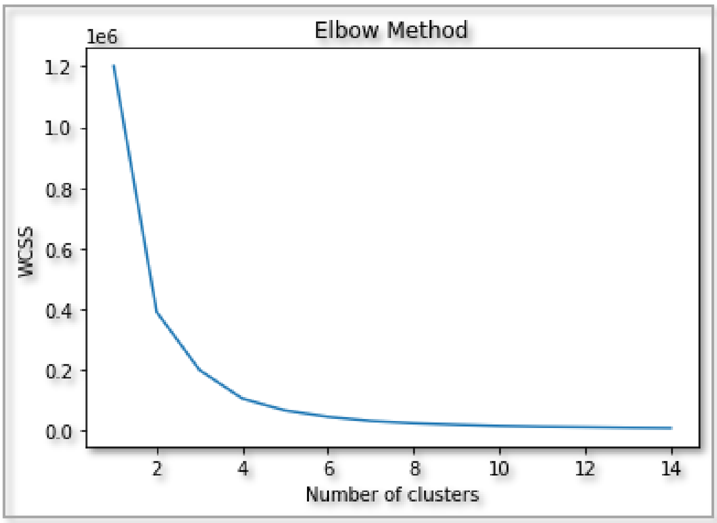
Additionally, the values in the columns “price” and “coefficient of location” were scaled from 0 to 1. This step is absolutely essential. Otherwise, the k-means clustering algorithm will give more emphasis on the variable that has higher absolute values (price), which consequently makes the algorithm biased. The end result of the scaled sample is the following data frame:

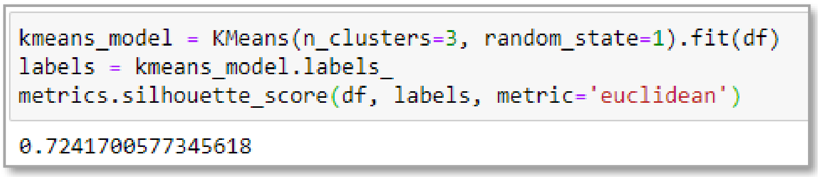


	price	coefficient_of_location	neighbourhood_group	code
196499	0.697318	0.263158	Amadora	1
196505	0.003831	0.263158	Amadora	1
196265	0.180077	0.263158	Amadora	1
196446	0.022989	0.456140	Amadora	1
196409	0.065134	0.263158	Amadora	1
...	...	...	...	...
193660	0.122605	0.368421	Oeiras	9
193819	0.141762	0.368421	Oeiras	9
193816	0.160920	0.368421	Oeiras	9
193886	0.149425	0.263158	Oeiras	9
193813	0.130268	0.368421	Oeiras	9

Figure 28: Overview of the Scaled Data

Now that the data is sampled and scaled, it is necessary to determine the ideal number of clusters that maximize the efficiency of the clustering algorithm. In order to achieve this, the Elbow Method was applied, followed by the Silhouette Coefficient:





```
kmeans_model = KMeans(n_clusters=3, random_state=1).fit(df)
labels = kmeans_model.labels_
metrics.silhouette_score(df, labels, metric='euclidean')
```

```
0.7241700577345618
```

Figure 29: Elbow Method and Silhouette Score

The Elbow Method graph clearly indicates that 3 clusters would be the best number of clusters for the data given in the sample. This can be seen in the figure above because the point whose x-value corresponds to 3 is the inflection point for the WCSS line. Besides that, the Silhouette Coefficient, which measures how similar an object is to its own cluster, is 0.724. This is a fairly high value, meaning that objects in the data seem to be well matched to their own cluster and poorly matched to neighboring clusters.

Before seeing the end result of the clustering algorithm, it is necessary to visualize how the municipalities are distributed. The municipality scatterplot can be seen below:

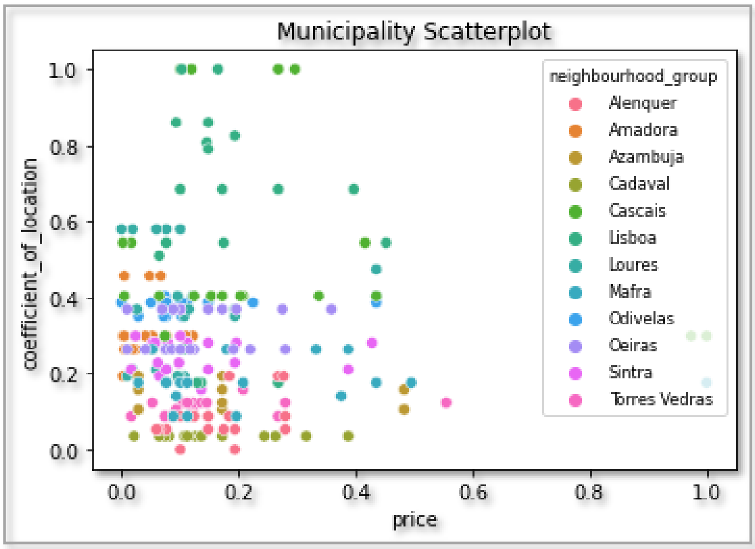


Figure 30: Municipality Scatterplot

After analyzing the above depicted graph, it is fair to conclude that, as expected, the clustering algorithm would never identify the 12 different municipalities simply because many of them are very alike. A good example of this is Cascais and Lisboa, which are distinct municipalities, but their coefficient of location does not differ much and, therefore, they are both very present on the top of the graph. On the other end of the spectrum, municipalities such as Torres Vedras, Cadaval, Azambuja and Alenquer, which are further away from the coastline and the center of Lisbon, have a much lower coefficient of location and cluster around the bottom-end of the graph.

Furthermore, it is possible to see that the price of the listings does not seem to be correlated with the coefficient of location at all. In fact, the majority of the expensive listings in this sample are present in municipalities with an overall low coefficient of location. Naturally, this happens because there are a lot more variables that influence the price other than the coefficient of location.

Lastly, all that is left to do is to define, fit and plot the k-mean clustering model. The end result of k-means clustering can be seen in the scatterplot below:

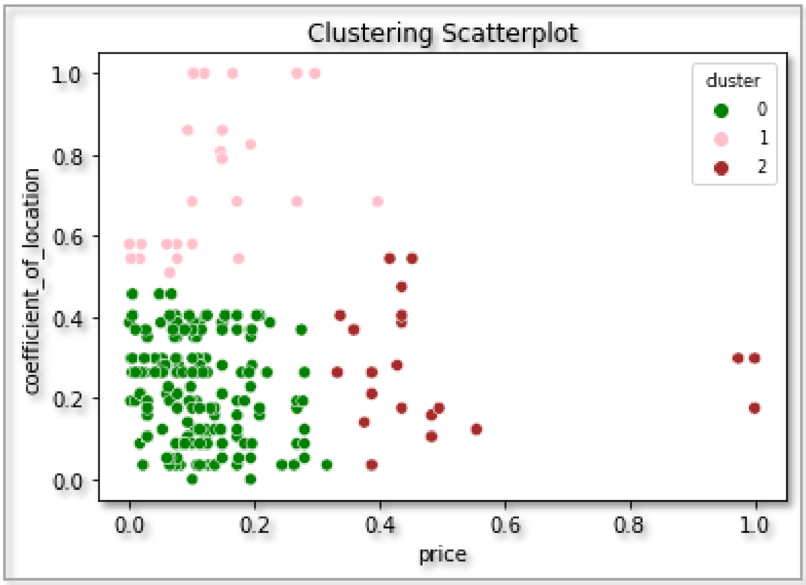


Figure 31: Clustering Scatterplot

The three clusters that were previously defined are now clearly depicted in the graph above. It seems that cluster 0 is the one with the most data points. This cluster is clearly identified as the one where data points have a low coefficient of location and a low price. Cluster 1 on the other hand, is composed of data points that have a low coefficient of location, but a higher average price than other clusters. Lastly, cluster 2 has data points with a high coefficient of location, but similar average prices to cluster 0.

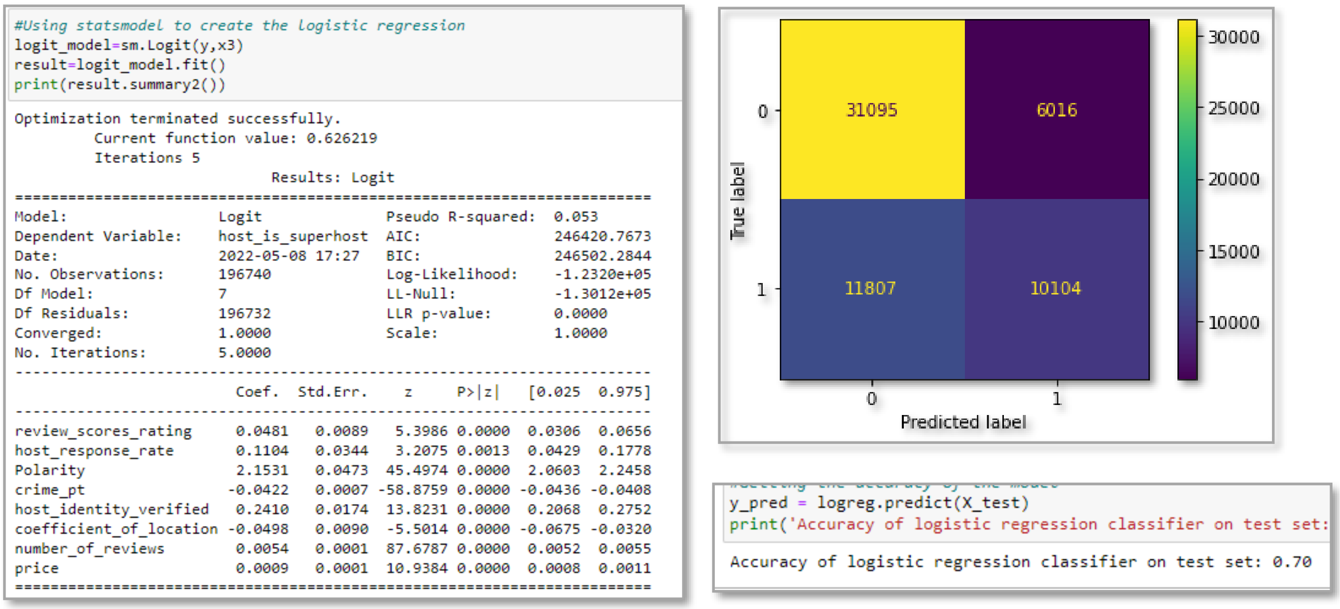
By comparing the clustering scatterplot with the municipality scatterplot, it is possible to identify which municipalities were clustered together. The two municipalities of Lisboa and Cascais tend to have data points with a higher coefficient of location and average prices and were therefore clustered together in cluster 2. On the other hand, the vast majority of the other municipalities have a lower coefficient of location and average prices. Hence the reason why they were nearly all clustered in cluster 0. Lastly, cluster 1 does not correspond to any specific group of municipalities, but rather a group of listings with a higher-than-average prices that are scattered throughout all the municipalities in the Airbnb listings.

To sum it up, by applying clustering to the given data, it was possible to conclude that, even though some municipalities seem to have some similarities, the vast majority of them have similar prices and location coefficients. Meaning that listings in towns with worse coefficients of locations (such as

Azambuja, Torres Vedras or Alenquer) don’t necessarily correspond to a lower pricing than listings in towns with better coefficients of locations (such as Lisboa or Cascais).

4.2.4 Classification Algorithms

In this section, the main objective was to identify which features and which algorithm would be the best to predict if the host was a super host or not. Being a super host clearly has a positive impact on the clients view, they tend to valorize and prefer hosts labeled as super host. Understanding the characteristics that clearly separate the difference between super hosts and hosts can be one of the keys to understand the features that turn accommodations more attractive. The first algorithm tested was the logistic regression.



```
#Using statsmodel to create the logistic regression
logit_model=sm.Logit(y,x3)
result=logit_model.fit()
print(result.summary2())
```

```
Optimization terminated successfully.
         Current function value: 0.626219
         Iterations 5
                             Results: Logit
=========================================================================
Model:                 Logit              Pseudo R-squared:   0.053
Dependent Variable:    host_is_superhost  AIC:                246420.7673
Date:                  2022-05-08 17:27   BIC:                246502.2844
No. Observations:      196740             Log-Likelihood:     -1.2320e+05
Df Model:              7                  LL-Null:            -1.3012e+05
Df Residuals:          196732             LLR p-value:        0.0000
Converged:             1.0000             Scale:              1.0000
No. Iterations:        5.0000
-------------------------------------------------------------------------
                         Coef.  Std.Err.     z     P>|z|   [0.025  0.975]
-------------------------------------------------------------------------
review_scores_rating     0.0481   0.0089   5.3986 0.0000  0.0306  0.0656
host_response_rate       0.1104   0.0344   3.2075 0.0013  0.0429  0.1778
Polarity                 2.1531   0.0473  45.4974 0.0000  2.0603  2.2458
crime_pt                -0.0422   0.0007 -58.8759 0.0000 -0.0436 -0.0408
host_identity_verified   0.2410   0.0174  13.8231 0.0000  0.2068  0.2752
coefficient_of_location -0.0498   0.0090  -5.5014 0.0000 -0.0675 -0.0320
number_of_reviews        0.0054   0.0001  87.6787 0.0000  0.0052  0.0055
price                    0.0009   0.0001  10.9384 0.0000  0.0008  0.0011
=========================================================================
```

```
y_pred = logreg.predict(X_test)
print('Accuracy of logistic regression classifier on test set:
```

```
Accuracy of logistic regression classifier on test set: 0.70
```

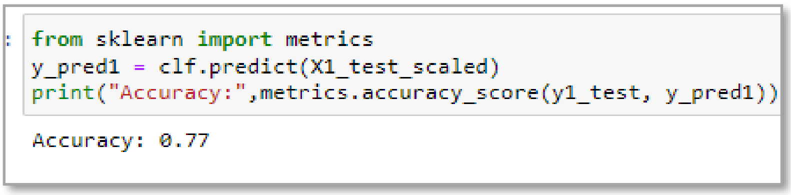
Figure 32: Logistic regression model

Multiple sets of features were tested and in the end it resulted in a logistic regression model with 8 features, including the features that Airbnb uses to classify if a host is super host or not. The result was a model that only explains 5.3% of the dependent variable variation which is relatively bad, however, it still gives us some very important interpretation about the features that other models can’t give.

All the coefficients are statistically significant, and the features that Airbnb uses to classify a super host are some of the ones that have more impact in the model (“reviews_score_rating”, “host_response_rate”). Yet, there are other features that stand out such as “host_identity_verified” and “Polarity” in a positive way and “crime_pt” and “coefficient_of_location” in a negative way. The price coefficient came out as almost a constant in the model, although it’s a good feature to use for classification, price is more a consequence of being a superhost then a reason to become one and that’s probably the reason why it’s almost zero.

Overall the model got an accuracy of 70% which isn’t that great. One of the main causes might be because there isn’t clear linear separable data. To make sure that the features truly have an impact on the superhost, the group decided to implement other algorithms that better suit the data to confirm our findings.

The next algorithm tested was the SVM classifier. Before implementing this algorithm, it was necessary to scale the data first, namely do a downsample, since the SVM aren’t suitable for large datasets.



```
from sklearn import metrics
y_pred1 = clf.predict(X1_test_scaled)
print("Accuracy:",metrics.accuracy_score(y1_test, y_pred1))

Accuracy: 0.77
```

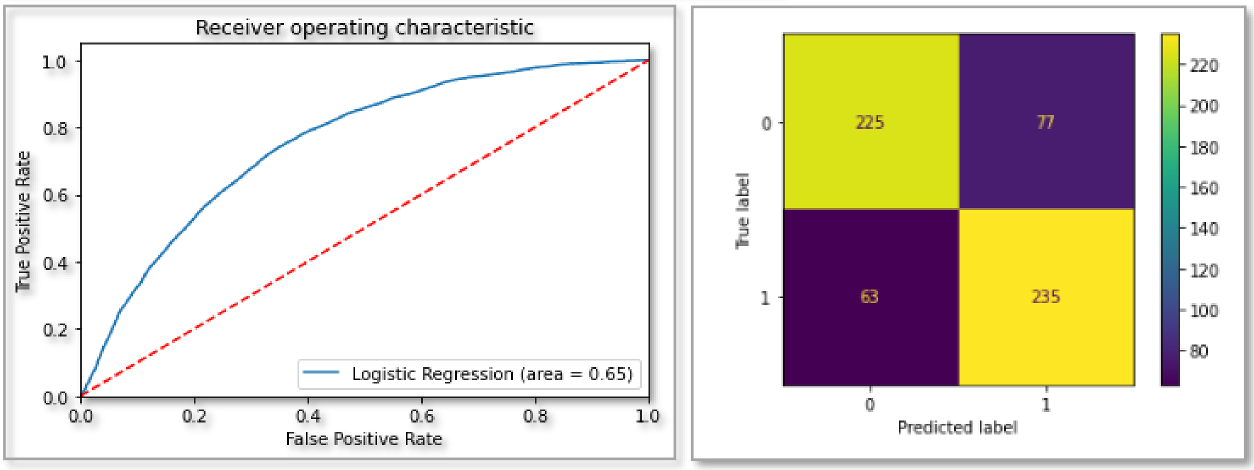
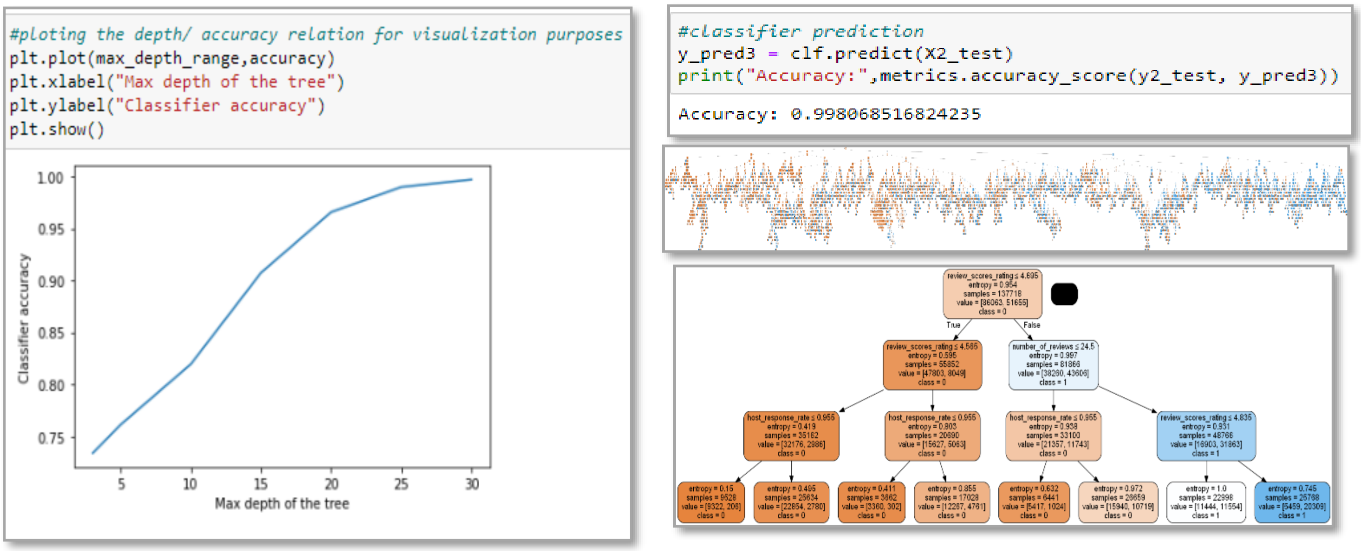


Figure 33: SVM model

The same features were selected for this algorithm. The model has a 77% a better accuracy than the logistic regression. However, it can’t give a statistical explanation about the classification and the model.

The last algorithm tested was the decision tree and random forest, the random forest was applied after the decision tree to ensure that the decision tree model wasn’t overfitting the data. The same features were used.



```
#ploting the depth/ accuracy relation for visualization purposes
plt.plot(max_depth_range,accuracy)
plt.xlabel("Max depth of the tree")
plt.ylabel("Classifier accuracy")
plt.show()
```

```
#classifier prediction
y_pred3 = clf.predict(X2_test)
print("Accuracy:",metrics.accuracy_score(y2_test, y_pred3))
```

```
Accuracy: 0.998068516824235
```

Figure 34: Classification tree model

This first graph shows that as the depth of the tree increases, the model becomes better. However, the model complexity also increases. Our model predicted the super host with 99.8% accuracy which is very good. To decrease the complexity of the model, the pruning method was applied. This made the model simpler and easier to visualize but decreased the accuracy of the model to 73%.

Lastly, the random forest method was implemented. It consisted on using cross validation with 10 folds and 3 repeats, which resulted and confirmed the accuracy of our decision tree (99.8%).

SVM and Decision tree models gave a good accuracy in predicting the superhost, so our features clearly impact if a host is a superhost or not. Moreover, these models allowed us to understand the main factors that a host has to focus on in order to become a superhost. Having a more attractive accommodation is related with the interaction between host and client ("host_response_rate", "reviews_score_rating", "polarity", "host_identity_verified") and it doesn't depend on the location or characteristics of the house. Building a good and healthy relationship with the costumers boosts the attractiveness of the accommodations.

5 Conclusion

During this project various methods in the area of descriptive statistics, but also algorithms in the area of inductive statistics were applied. This has allowed the group to gain important insights into Airbnb tourism in Lisbon and to answer the underlying research question: What are the factors that make an Airbnb accommodation more attractive?

In summary, the main conclusions are the following:

- The status as a superhost can be a factor of influence that makes an accommodation more attractive and hence more expensive.
- Factors, such as a high number of bedrooms/bathrooms, a high review score or a good polarity score of comments, can justify higher prices of accommodations.
- Listings in towns with worse coefficients of locations don't necessarily correspond to a lower pricing than listings in towns with better coefficients of locations.
- Building a good and healthy relationship with the costumers boosts the attractiveness of the accommodations.
- The most successful host in terms of numbers of apartments offered on Airbnb offered focused on the centrality of the accommodation.

However, there are some limitations in the study. For example, a higher R-squared of the multivariate linear regression and logistic models would be desirable. Additional data which could explain the variations in price should be explored in further research on this topic. Furthermore, the group didn't manage to translate the comments of the reviews to a unique language, and this has a direct effect on polarity analysis since it's based on the English language only.

Moreover, it was assumed that the OLS assumptions were fulfilled. No model specification tests were applied. Therefore, we propose that the results presented within this study should not be generalized on an international level without further research.

Nevertheless, we believe that those findings can help hosts to increase the attractiveness of their accommodations, but also Airbnb since the company's revenues model is based on commissions from successful bookings.

6 References

Alberto Amore, Cecilia de Bernardi & Pavlos Arvanitis (2020) “The impacts of Airbnb in Athens, Lisbon and Milan: a rent gap theory perspective”, Current Issues in Tourism, DOI: 10.1080/13683500.2020.1742674

Analytics Vidhya (2021) “End-to-End Predictive Analysis on AirBnB Listings Data, accessed on 06. May 2022, https://www.analyticsvidhya.com/blog/2021/10/end-to-end-predictive-analysis-on-airbnb-listings-data/#h2_8.

Business Model Toolbox (2020) “Airbnb”, accessed on 28. April 2022, https://bmtoolbox.net/stories/airbnb/.

C. J. Costa and J. T. Aparicio (2020) "POST-DS: A Methodology to Boost Data Science," 15th Iberian Conference on Information Systems and Technologies (CISTI), Seville, Spain, 2020, pp. 1-6, doi: 10.23919/CISTI49556.2020.9140932.

Haklay, M. and Weber, P. (2008) ‘OpenStreetMap: User-Generated Street Maps’, IEEE Pervasive Computing, 7(4), pp. 12–18. doi:10.1109/MPRV.2008.80.

Inside Airbnb (2022) “Get the Data”, accessed on 28. April 2022, http://insideairbnb.com/get-the-data.

McKnight, P.E. and Najab, J. (2010) ‘Mann-Whitney U Test’, The Corsini Encyclopedia of Psychology [Preprint]. doi:https://doi.org/10.1002/9780470479216.corpsy0524.

Pordata (2020) “Purchasing Power per Capita”, accessed on the 28th of April 2022, https://www.pordata.pt/en/Municipalities/Purchasing+power+per+capita-118.

Pordata (2020) “Population Density by Municipality”, accessed on the 28th of April 2022, https://www.pordata.pt/en/Municipalities/Population+density-452.

Pordata (2020) “Crimes Registered per Thousand Inhabitants” accessed on the 28th of April 2022, https://www.pordata.pt/en/Municipalities/Crimes+registered+by+police+per+thousand+inhabitants-995.

Towards Data Science (2019) “Airbnb Rental Listings Dataset Mining”, accessed on 06. May 2022, https://towardsdatascience.com/airbnb-rental-listings-dataset-mining-f972ed08ddec.

Tribloom (2019) “CRISP-DM, one AI/ML Lifecycle”, accessed on 28. April 2022, https://www.tribloom.com/crisp-dm-one-ai-ml-lifecycle/.

Wu, Q. (2021) ‘Leafmap: A Python package for interactive mapping and geospatial analysis with minimal coding in a Jupyter environment’, Journal of Open Source Software, 6(63), p. 3414. doi:10.21105/joss.03414.